# Getting Started in DIAMOND

DISCLAIMER: The authors of DIAMOND make no warranty, implied or expressed, that DIAMOND is in any way free of errors or bugs. Also, DIAMOND is provided as freeware for all to use, distribute, and modify. No technical support can be provided, so please do not ask for any.

## *Loading Data*

### Loading UFF Data

Choose File->Import->Universal->Type 58

Select the file

Save as DIAMOND Data file using Load/Save -> Save -> Data

### Loading MATLAB Data

To load a DIAMOND format data file, use Load/Save -> Load -> Data

If you have data in MATLAB variables that do not conform to the DIAMOND file structure, it is probably easiest to write a simple MATLAB script to convert the data from your data structure to the DIAMOND data structure.

## *Defining Geometry*

Select menu: Geometry/DOF -> Edit Geometry

For viewing mode shapes, must define nodes and tracelines

For doing node-based damage indicators (e.g. flexibility) and other methods that use connectivity information, must define bars and shells

For strain energy methods, must define SEMBeams and SEMQuads

## *Plotting Modal Data*

Select menu: Plot Data -> Plot Spectral Data

## *Identifying Modal Parameters*

Select method under menu: Identify Modes

Only operating shape and rational polynomial methods are even somewhat reliable

### *Saving Results*

Select menu: Load/Save -> Save -> Data or Geometry or Modes

## DIAMOND DEMONSTRATION FILES

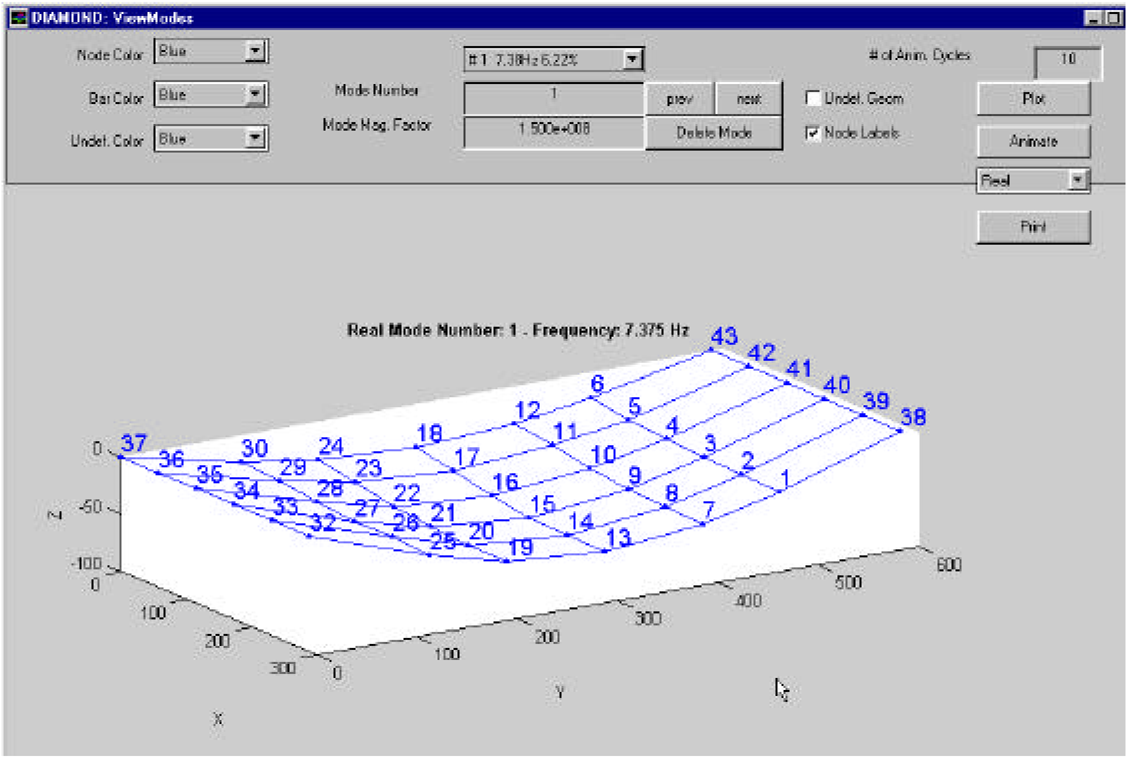
The following demonstrations can be performed using the mat-files included with the DIAMOND software in the folder ‘diamond_demo’. Sample screenshots are included to show approximately what you should be seeing for each case.

### *Experimental Modal Analysis*

This example will demonstrate how to extract modal frequencies, modal damping, and mode shapes from frequency response function data using the operating shape, rational polynomial, and ERA techniques. Data from the Alamosa Canyon modal test series will be used.

- Start DIAMOND from the matlab command prompt, and select the ‘start’ button after DIAMOND has initialized.
- Load the data:
    - Load/Save → Load → Data → Primary
    - Select file ‘diamond_demo/acbt/acb_data_hammer.mat’
- Load the Geometry
    - Load/Save → Load → Geometry
    - Select file ‘diamond_demo/acbt/acb_geom.mat’
- Perform the identification
    - Option 1: Operating Shapes (“Peak-Picking”)
        - Identify Modes → Operating Shapes
        - Click “Use Mouse” button, when crosshair appears, select first modal peak in data (about 7.375 Hz). If you miss the peak, use the arrows at the bottom of the screen to tweak the location of the cursor.
        - Click “Define Op Shape” button. The Viewmodes window will appear with the geometry (viewed from above). Grab near node 32 with the mouse and drag up and to the left a couple of inches, then release the button. You should

now have an isometric view of the first mode shape (actually operating shape) that looks like this:



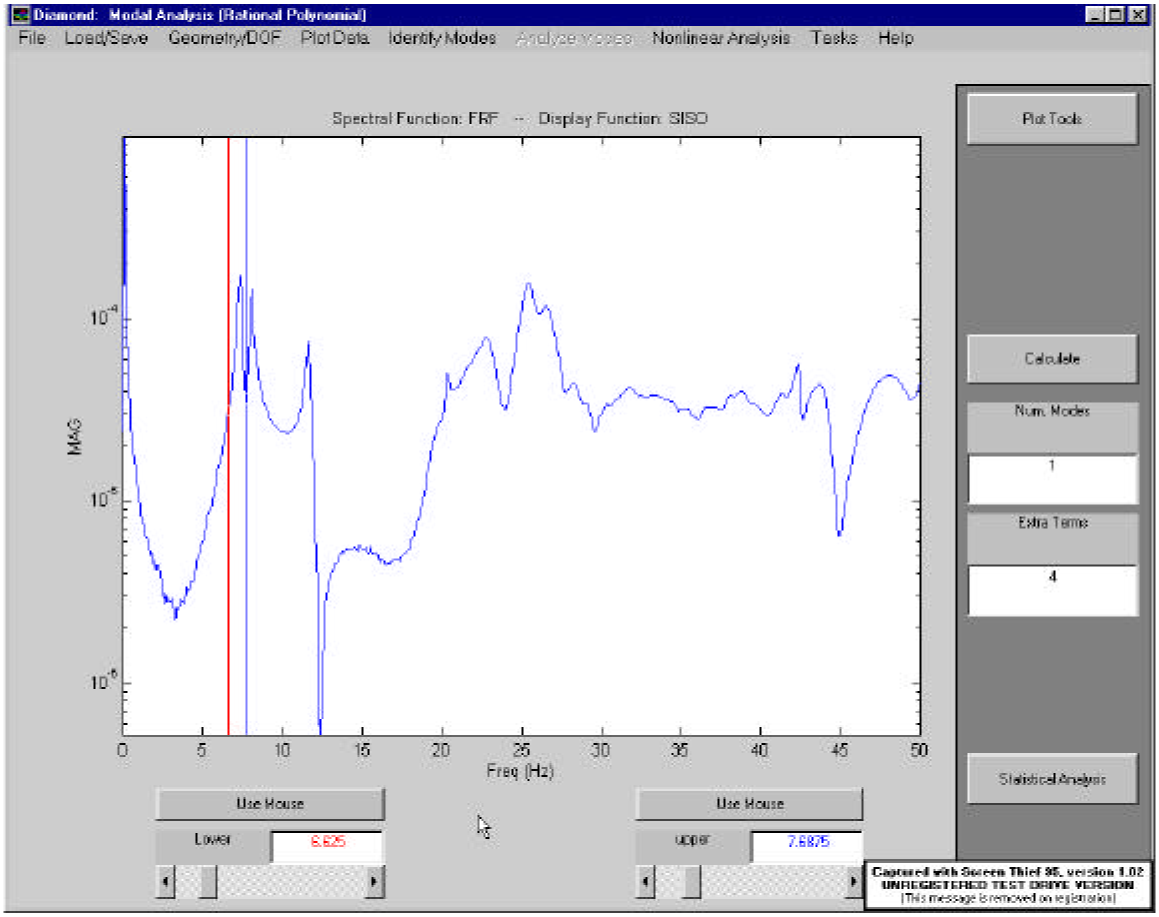

- Click the “exit viewmodes” button to close the viewmodes window.

- Select Analyze Modes → Clear Modes to clear the operating shape that you just defined.

- Option 2: Rational Polynomial Fit

  - Identify Modes → Rational Polynomial

  - Position the cursors at about 6.6 Hz and 7.7 Hz respectively using the cursor controls at the bottom of the screen. The cursors should bracket the first modal peak like this:



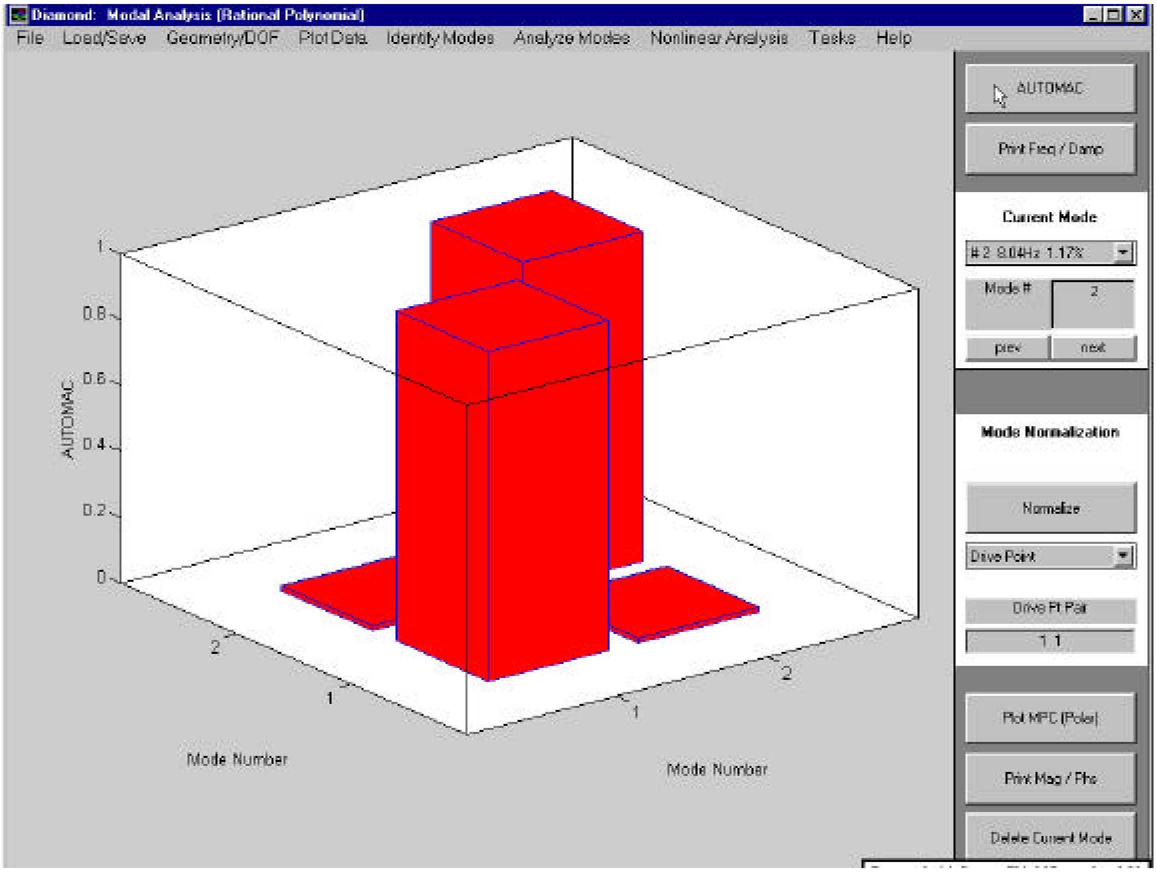

- Set Num Modes = 1 and Extra Terms = 4, and click "calculate". The viewmodes window should appear and show the first mode shape. This first mode shape should be quite similar to the first operating shape shown above.

- Repeat the procedure for the second peak. You should now have two modes defined at about 7.35 Hz and 8.03 Hz.

- Post-Processing the modes.

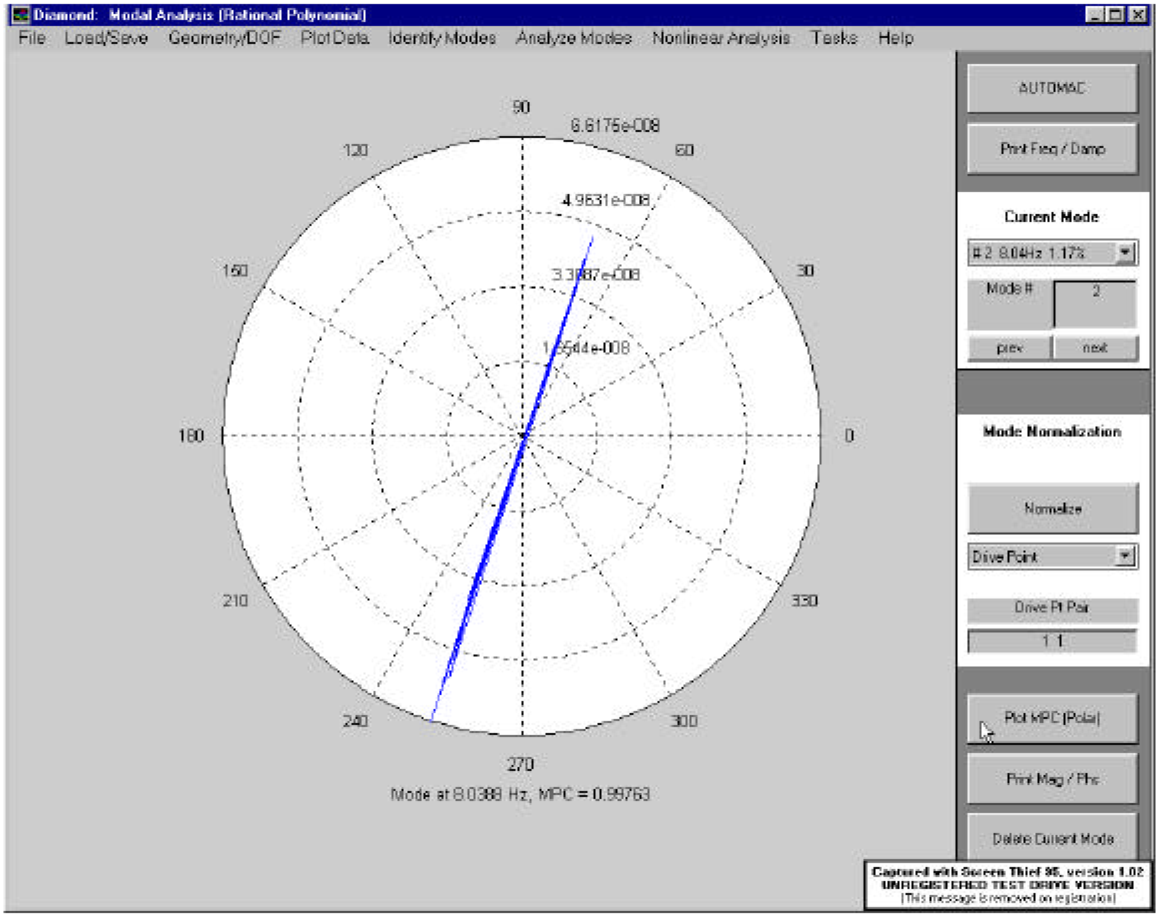
  - With the two modes previously defined in memory, select Analyze Modes → Analysis Tools

  - Click the "AUTOMAC" button. A 3-D bar chart of the modal assurance criteria (MAC) of these two modes with respect to themselves is shown. It should look like this




- Click the "Print Freq/Damp" button. A table of the modal frequencies and damping ratios currently in memory is displayed in the Matlab command window. This table is suitable for printing.

- Click the "Plot MPC (Polar)" button. This produces a plot of the modal phase collinearity (MPC) for the current mode. This plot is the magnitude vs. phase for each component of the complex modal vector. It is an indication of the level of 'normality' of the identified mode shapes. For a normal mode shape, this plot should appear as a nearly straight line. For the second mode identified above, the MPC should look like




- Clicking the "Print Mag/Phs" button prints a table of the magnitudes and phases of the current mode in the Matlab command window. These should be the same values that are seen in the MPC plot, but may be rotated by a constant angular amount.

### *Modal-based Damage Identification*

This example will demonstrate the use of modal-based damage identification techniques to local damage in a structure. Data from the I-40 modal test series will be used. For reports and photographs on the I-40 tests, please see

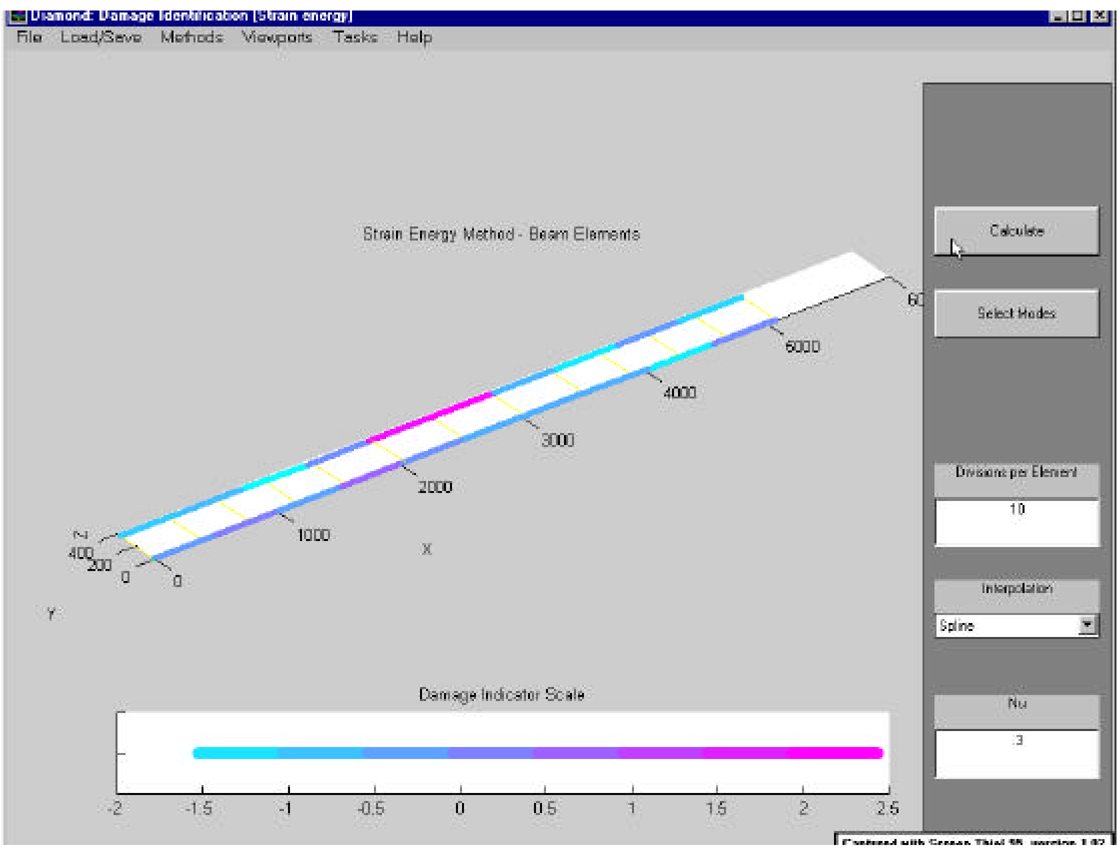
http://esaea-www.esa.lanl.gov/damage_id/

- From the Tasks menu, select Tasks → Damage Identification (NOTE: To avoid problems, it is usually desirable to quit and restart MATLAB and DIAMOND prior to switching to a new Task)

- Load the undamaged modal data:

  - Select Load/Save → Load Undamaged Modes

- Select file ‘diamond_demo/i40/dam0_modes.mat’
- Load the damaged modal data (damage case 4)
  - Select Load/Save → Load Damaged Modes
  - Select file ‘diamond_demo/i40/dam4_modes.mat’
- Load the geometry
  - Select Load/Save → Load Geometry
  - Select file ‘diamond_demo/i40/i40_geom.mat’
- Apply the modal damage identification technique
  - Option 1: Strain Energy Methods
    - Select Methods → Strain Energy Methods
    - Click “Select Modes”, check “Use All modes”, click “close”
    - Set “Divisions per Element” = 10 and click “calculate. Results should look like:



- Notice that the pink area (highest indicator level) is around the point of the damage (Node 20, x=2500, y=400)

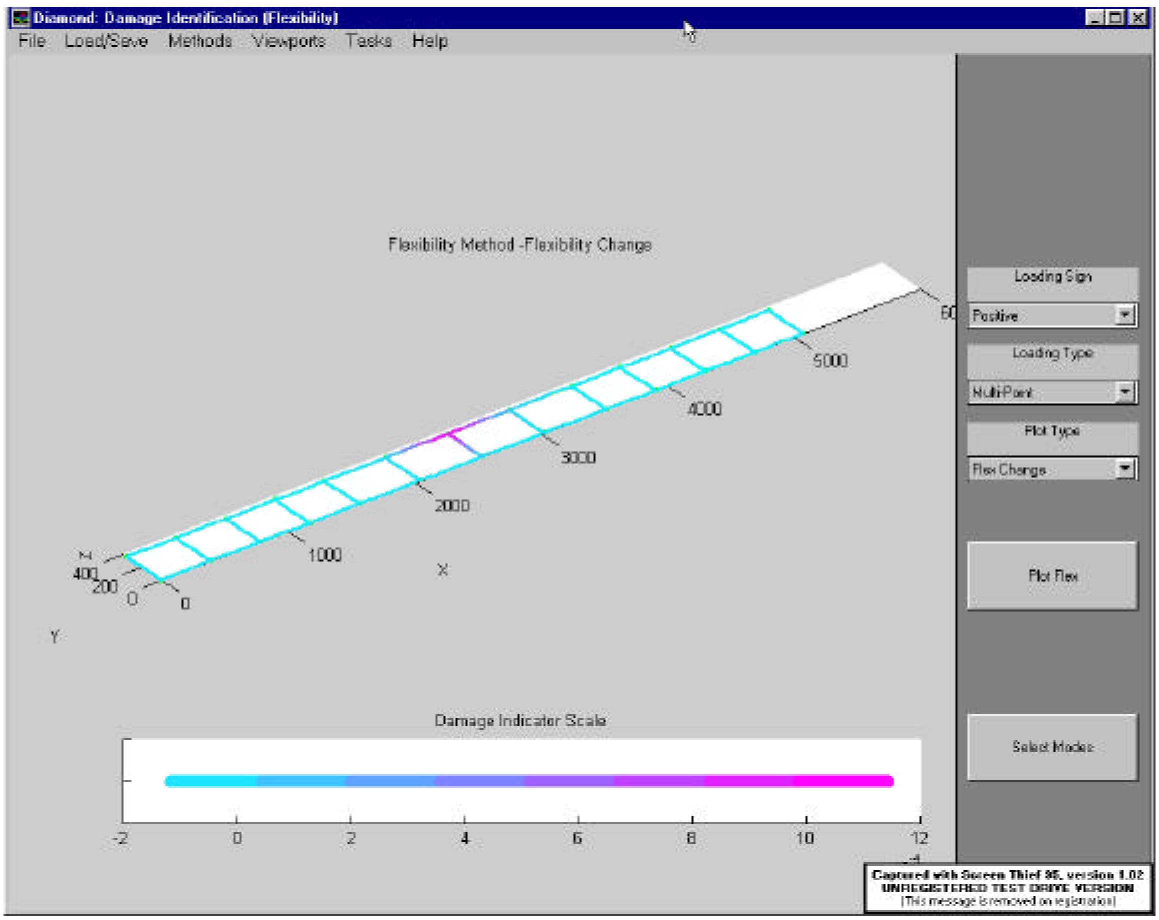
- Change the number of divisions per element to 20 and re-calculate the damage indicator. Notice that the location of the damage is now more specific spatially. Change it to 40 and notice the continued refinement, but also the increase in computation time.

- Option 2: Flexibility methods

  - Select Methods → Flexibility Methods

  - Click "Select Modes", check "Use All modes", click "close"

  - Set Loading Sign = Positive, Loading Type = Multi-Point, Plot Type = Flex Change. Click Plot Flex. The results should look like this:




- Option 3: MAC/COMAC Method

  - Select Methods → MAC/COMAC

- Click the button “Damage MAC”. This gives the modal assurance criteria (MAC) between the undamaged and damaged data sets. This plot can give insight as to which modes exhibit the largest change as a result of damage. This knowledge is helpful in selecting which modes to use for the damage identification techniques.

- Click the button “Damage COMAC,” and you should see an overlay of the coordinate MAC (COMAC) on the structural geometry. The COMAC is similar to the MAC, but is spatially referenced to each degree of freedom. A lower COMAC indicates areas of discrepancy between the undamaged and damaged mode sets. In this case, the lowest values of COMAC appear at the center of the middle bridge span, which is near the damage location.

### *Model Update-based Damage Identification*

This example will demonstrate the use of finite element model updating techniques to locate damage in a structure. Data simulated from a finite element model of a rectangular plate will be used.

- From the Tasks menu, select Tasks → Model Correlation (NOTE: To avoid problems, it is usually desirable to quit and restart MATLAB and DIAMOND prior to switching to a new Task)

- Load the (undamaged) model modes

    - Select Load/Save → Load Model Modes

    - Select the file ‘diamond_demo/plate_demo/udmofull.mat’

- Load the measured (damaged) modes

    - Select Load/Save → Load Measured Modes

    - Select the file ‘diamond_demo/plate_demo/dmmofull.mat’

- Load the (undamaged) model matrices

    - Select Load/Save → Load Model Matrices

    - Select the file ‘diamond_demo/plate_demo/udmtfull.mat’

- Load the sensor geometry

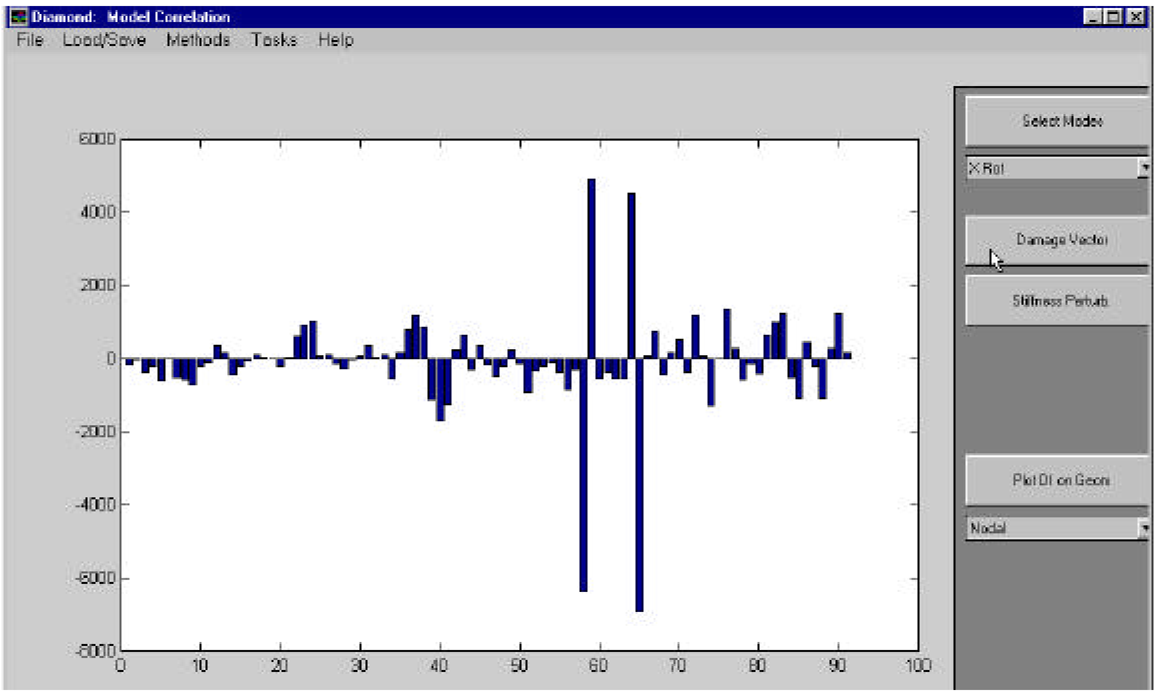
    - Select Load/Save → Load Geometry

    - Select the file ‘diamond_demo/plate_demo/plate_geom.mat’

- Select the Minimum Rank Perturbation Technique (MRPT)

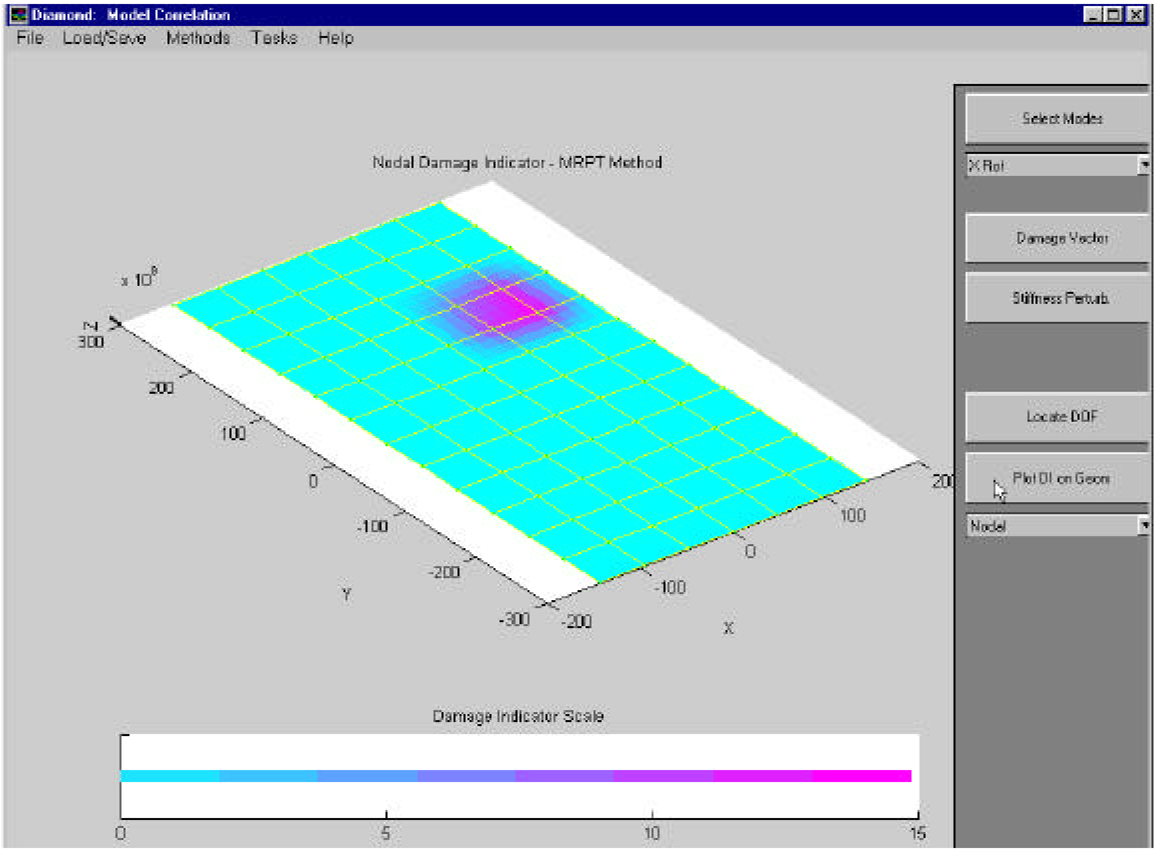
- Select Method → Min. Rank Perturbation (MRPT)

- Examine the modal force error vector
  - Click “select modes” and check “use all modes”. Click “close”
  - Select “X rot” from the pull-down menu
  - Click “damage vector” to compute the modal force error. The resulting plot should look like the following figure. This plot clearly shows that the damage is localized around DOF 58, 59, 64 and 65.




- Examine the change in the stiffness matrix
  - Click the “stiffness perturb.” Button to compute the stiffness matrix perturbation.

- Click “Plot DI on Geom” to plot the damage indicator on the geometry. The resulting window should look like the following figure. Notice that the damage is well identified in one of the quadrilateral elements.




## Test Simulation

This example will demonstrate the use of the test simulator module.

- From the Tasks menu, select Tasks → Test Simulation (NOTE: To avoid problems, it is usually desirable to quit and restart MATLAB and DIAMOND prior to switching to a new Task)

### *Defining the system.*

The first step in utilizing the test simulation portion of DIAMOND is to define the system to be simulated. This is done by selecting "System | Define / Modify". A dialog window then appears.

There are two types of systems that can be simulated. The first is a system with arbitrary mass, stiffness and damping matrices. If this alternative is chosen, the user has the option of specifying and arbitrary damping matrix or specifying modal damping values and calculating the resulting damping matrix.

The second type of system is a chain of masses connected together with springs and dampers to ground. Thus, if this option is chosen, the user is expected to enter a vector of masses, stiffnesses, and damping values, starting with the elements closest to ground. The user is also given the option of base excitation. If this is selected, the input is taken to be an acceleration, rather than a force. Thus, the output will also be an acceleration. Notice that if the system is specified as a chain of masses, the modal damping option is not available. This is because, in general, the resulting damping matrix would not be consistent with a simple chain of masses. If the user *really* wants to use modal damping in this case anyway, the following procedure may be followed.

1. Enter the mass and stiffness vectors.
2. Enter the damping vector as just zeros.
3. Press the OK button.
4. Select "System | Define / Modify" again.
5. Now select the modal damping option.
6. Enter the vector of modal damping coefficients.
7. Press the OK button.

### *Defining the input.*

Once the system has been defined, the input needs to be established. For this purpose, several types of input generation are available. Note that it is assumed that there exists only one input to the system.

#### Random input signal

This option allow the user to create a random input signal with a specified power spectral density. The user supplied PSD consists of a matrix with two columns and as many rows as are necessary to describe the desired PSD. That is, each row represents a frequency-magnitude pair. Therefore, the first column (the frequency column) must be monotonic. In addition, the frequencies in the PSD definition can not exceed the nyquist frequency of the input signal, or an error will result.

#### Harmonic input signal

This option allows the user to create a signal which consists of a sum of harmonics of various frequencies. Thus the number of components is the number of frequencies which will be included in the signal. The vector of frequencies contains as many elements as there are components. The elements of the vectors of amplitudes, start times and stop times, are the amplitude, start time and stop time for each component, respectively. Note that the actual stop times may be slightly different than requested since the program stops at the nearest half-cycle in an effort to minimize discontinuities.

### Impulse input signal

This option allows the user to create a half sine wave impulse of specified duration and amplitude. The start time for the impulse is also specified by the user. It should be noted that the resulting input signal is spaced uniformly in time. Thus, for a long signal consisting of a very brief impulse, a large number of points should be used in order to adequately represent the impulse.

### User defined input signal

This option allows the user to enter an input of their choice along with the corresponding time vector. For simulation purposes, the time vector must be uniformly spaced.

## *Defining a window*

Once the input signal has been chosen, the user has the option of applying a window to the input. The choices for a window are a hanning, exponential, rectangular, or user defined. For the exponential window, the "alpha" parameter which the user is prompted for is the value of the window (between 0 and 1) at the stop time for the window.

Once the simulation has been run, this same windowing option is available for the output. Note, however, that the window selected will be applied to all outputs.

## *Running the simulation*

Once the input has been defined and windowed as desired, the simulation may be run. At this point, the only information that the user needs to supply is the initial conditions (position and velocity) of the system. Please note that if a window of the input is desired, it must be defined each time before the simulation is run. This is to prevent windows of windows.

## *Plotting results*

Similar to running the simulation, if a window of the output is desired, it must be defined each time before the results are plotted. Again, this is to avoid windows of windows.

## *Sending results to the workspace*

If the user wishes to manipulate the output of the system further, the results may be sent to the workspace as variables of the user's choice. Note that the results must be plotted before they can be sent to the workspace.

## *A word about Loading and Saving*

At any point in the simulation procedure, everything which has been entered may be saved. Thus, if the user were to close Matlab and restart it, then by loading the system model, the system along with any inputs defined or outputs calculated would be immediately available.

## An illustrative example

The following step should illustrate the test simulation portion of DIAMOND with adequate clarity.

1) Start DIAMOND
2) Select “Tasks | Test Simulation”
3) Select “System | Define / Modify”
4) In the popup menu, select chain of masses connected to ground
5) For the mass vector, enter [3 5 7]
6) For the stiffness vector, enter [6 9 11]
7) For the damping vector, enter [0 0 0]
8) For the forcing vector, enter [0 1 0]
9) Click “OK”
10) Select “Tasks | Test Simulation”
11) Select the modal damping option
12) For the zeta vector, enter [.2 .4 .3];
13) Click “OK”

We have now defined a system of 3 masses connected to ground. Even though it may not make any physical sense, we have chosen to have modal damping. Now we are ready to define an input. For the purposes of this example, we generate an impulse input.

14) Select “Simulation | Input Signal | Impulse”
15) Enter the following parameters in the text boxes
    a) “Total signal duration”: 20

b) “Number of points”: 1000

c) “Pulse start time”: 1

d) “Pulse duration”: 0.2

e) “Pulse amplitude”: 15

16) Click “OK”

Now that we have our input signal, let’s window it with a hanning window.

17) Select “Simulation | Windowing | Hanning”

18) Enter 0 for the start time, and 20 for the stop time.

Now we are ready to run the simulation.

19) Select “Simulation | Run”

20) Enter [0 0 0] for the initial position and velocity

21) Click “OK”

Now we are ready to look at the results

22) Select “Post Processing | Plot Results”

Notice that the input signal has been windowed. Since this was an impulse signal, the window did little more than decrease the amplitude of the impulse. Lets run the simulation again without the window.

23) Select “Simulation | Run”

24) Make sure the initial conditions are still zero and click “OK”

Now lets plot the results with and without an output window.

25) Select “Post Processing | Plot results”

26) Select “Post Processing | Window Results | Rectangular”

27) Enter 3 for the start time and 15 for the stop time; click “OK”

28) Select “Post Processing | Plot results”

Compare the two plots to verify the windowing action. Now let’s send the output to the workspace. Note that the windowed output will be sent since it was the most recent thing we plotted.

29) Select “Post Processing | Output Results to Workspace”

30) Enter “t” for the time vector and “y” for the output matrix”

Go to the command window enter “plot(t,y)” and verify that this is the same plot as the windowed output plot.